REMARKS/ARGUMENTS

Reconsideration of the application, in view of the following remarks is respectfully requested.

The Examiner rejects Claims 12, 15-21 under 35 U.S.C. 103(a) as being unpatentable over Kelly et al. in view of Hung et al. The Examiner states that Kelly disclosed an arbitration circuit for an output port comprising a FIFO queue containing a head pointer and the Examiner specifically refers to column 4, lines 50-67 to store in a common memory for a plurality of ports and plurality of characterizing data for each packet received at an input port in the data portion of the packet is stored in the common memory.

This rejection is respectfully traversed. First of all, the Examiner has ignored those statements made in our previous responses which showed that the statements made by the Examiner characterizing Kelly et al. are incorrect. The Examiner has substituted Hung et al for Chin et al to address the distinction we made concerning the terms "header" and "head pointer". Assuming, arguendo, that Hung et al addresses this point, this does not address the other distinctions that we made over Kelly et al, which the Examiner has repeatedly ignored. In addition, as will be discussed herein below, Hung et al further teaches away from the present invention.

As previously stated, Kelly does not store packets in a common memory, as stated by the Examiner. In a previous response we quoted from Kelly, column 8, lines 40-65 which we now repeat below:

"Only a single input buffer set 831 is provided for handling traffic, per virtual channel, in the downstream path from the upstream port. Separate single input buffer sets 839 and 841 are provided at each of the downstream ports, per virtual channel for handling traffic in the upstream direction." (emphasis added)

Thus, as we previously stated, it is clear that Kelly et al. functions in terms of the prior art by requiring separate memories for the input port and for each of the two output ports. Furthermore, each of these memories has a separate buffer per virtual channel.

The Examiner states that Kelly shows a plurality of arbitration circuits coupled to the look-up table for selecting the next packet to be sent out corresponding to a preselected characterizing datum and she refers to column 9, lines 1-48.

However, referring to column 9 of Kelly et al., referred to by the Examiner, specifically lines 10-15, which recites:

"...to allow for steering transactions flowing out of the input buffers toward target buffers at the appropriate target output ports. This non-blocking switch 833 allows transfers to occur between any two combination of ports of the switch while simultaneously allowing transfers to occur between any two other combinations of sets of two ports of the switch" (emphasis added).

It is clear that the information is stored in input buffers and then transferred by the switch to buffers in the target output ports, thus negating any idea of a common memory to be used both by the input ports and the output ports, as in the present invention. The Examiner states that the x-bar switch is a common memory. However, those skilled in the art understand that a x-bar (crossbar) switch is just that –a switch, not a memory. See the enclosed definitions.

Furthermore, referring again to column 9, at lines 54-58, it recites:

"Managing only one input buffer set and only one output buffer set (per port per virtual channel) with improved ordering requirements is much less complex than managing multiple input and output buffer sets (per port per virtual channel) which have a much more complex ordering..." (emphasis added).

Thus, it is clear from Kelly et al. itself, as recited in two portions of column 9, specifically referred to by the Examiner, that Kelly et al. functions completely differently than the Examiner has characterized it and teaches away from utilization of a single buffer for all the input ports and output ports to share, with only the head pointers being transferred.

The Examiner states that Kelly et al. discloses all of the limitations as above but does not explicitly disclose a FIFO queue containing a head pointer to store in a common memory for plurality of ports. The Examiner states that Hung et al. discloses a head pointer for a port points to the main queue that contains a pointer to a packet that is to be next transmitted from the port. She therefore concludes it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate Hung's teaching into the Kelly system so as to provide a an increasing efficiency network.

This rejection is respectfully traversed. Firstly, Kelly et al teaches away from the utilization of a single memory shared by all the ports of the device. See the above discussion. So does Hung et al. In Hung et al, at column 3, lines 3-5 and 13-17, it recites:

" ...FIFO 310 and FIFO 330 maintain a separate storage location corresponding to each port coupled to MAC 210 and GMAC 215, respectively." (emphasis added) and

"Each of the FIFO buffers within the transmitter 235 store received data packets before the packets are forwarded to MAC 220 or GMAC 225 where they are transmitted from a network port." (emphasis added).

Kelly et al states that it would require a separate buffer for each port per virtual channel, and since it is well known that PCI Express supports eight virtual channels, Kelly would require a minimum of 24 separate memories for the configuration discussed in Figure 8, whereas the present invention requires only a single memory. Combining Kelly et al. with Hung et al. would not change that result.

Therefore, combining Kelley et al which teaches away from the present invention with Hung et al which teaches away from the present invention can not result in the present invention being obvious excerpt in impermissible hindsight after reading Applicants disclosure.

Accordingly, Applicants believe that the application, as previously amended, is in condition for allowance, and such action is respectfully requested.

Respectfully submitted,

/William B. Kempler/

William B. Kempler
Reg. No. 28,228
Texas Instruments Incorporated
PO Box 655474, M/S 3999
Dallas, Texas 75265
972.917.5452

Crossbar switch

From Wikipedia, the free encyclopedia

A **crossbar switch** (also known as **cross-point switch**, **crosspoint switch**, or **matrix switch**) is a switch connecting multiple inputs to multiple outputs in a matrix manner. Originally the term was used literally, for a matrix switch controlled by a grid of crossing metal bars, and later was broadened to matrix switches in general. It is one of the principal switch architectures, together with a memory switch and a crossover switch.

Contents

- 1 General properties
- 2 Applications
- 3 Implementations
  - 3.1 Mechanical
  - 3.2 Electromechanical / telephony
  - 3.3 Telephone exchange
  - 3.4 Semiconductor
- 4 Arbitration
- 5 See also
- 6 References
- 7 External links

General properties

A crossbar switch has a characteristic matrix of switches between the inputs and the outputs. If the switch has M inputs and N outputs, then a crossbar has a matrix with M x N cross-points or places where the "bars" cross. A given crossbar is a single layer, non-blocking switch. Collections of crossbars can be used to implement multiple layer and/or blocking switches.

Applications

Crossbar switches are most famously used in information processing applications such as telephony and packet switching, but they are also used in applications such as mechanical sorting machines with inputs. The crossbar "format" is also used in some semiconductor memory devices (see nanotechnology). Here the "bars" are extremely thin metal wires, and the "switches" are fusible links. The fuses are blown or opened using high voltage and read using low voltage. Such devices are called Programmable read-only memory. [1] At the 2008 NSTI Nanotechnology Conference a paper was presented which discussed a nanoscale crossbar implementation of an adding circuit used as an alternative to logic gates for computation.[2]

Implementations

Historically, a crossbar switch consisted of metal bars associated with each input and output, controlling movable contacts at each cross-point. In the later part of the 20th Century these literal crossbar switches

SearchNetworking.com Definitions (Powered by WhatIs.com)

LOOK UP TECH TERMS Powered by:

Search listings for thousands of IT terms: SEARCH

Browse tech terms alphabetically:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z #

cross-bar switch

DEFINITION - In a network, a cross-bar switch is a device that is capable of channeling data between any two devices that are attached to it up to its maximum number of ports. The paths set up between devices can be fixed for some duration or changed when desired and each device-to-device path (going through the switch) is usually fixed for some period.

Cross-bar topology can be contrasted with bus topology, an arrangement in which there is only one path that all devices share. Traditionally, computers have been connected to storage devices with a large bus. A major advantage of cross-bar switching is that, as the traffic between any two devices increases, it does not affect traffic between other devices. In addition to offering more flexibility, a cross-bar switch environment offers greater scalability than a bus environment.

In an IBM mainframe environment, the ESCON director is an example of a cross-bar switch.

LAST UPDATED: 17 Apr 2007